LISTING OF CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Original) A method in a computerized device for maintaining a client session in a network having a plurality of routers, the network having an application executed at a plurality of replicas, comprising the steps of:

provid ing a database of bindings of request identifiers to replicas where each binding is a record having a request identifier, a replica identifier and a binding expiration time, the database associated with a first router of the plurality of routers;

maintaining a change log of records entered into the database, each change log entry having a change event generated by the first router and an event number sequential to an event number of a preceding change event in the change log;

maintaining a current version vector associated with the database and the change log, the current version vector entry for the first router being a most recent event number from the change log, the current version vector entry for each other router being a most recent event number received at the first router from that other router;

receiving an update of change events generated at another router in the plurality;

reconciling the current version vector according to the received update; and

reconciling the database according to the received update such that the client session is maintained.

2. (Original) The method of claim 1 wherein the request identifier is a client identifier and an application identifier.

3. (Original) The method of claim 2 wherein the client identifier is an Internet Protocol address.

4. (Original) The method of claim 2 wherein the client identifier is a dproxy Internet Protocol address such that the binding associates a dproxy with a replica.

5.   (Original) The method of claim 1 wherein the step of reconciling the current version vector comprises the steps of:

comparing a least recent event number of the router that generated the update to the event number in the current version vector entry for that router;

if the least recent event number is in series with the event numbers in the database as determined by the current version vector entry for that other router, then entering the most recent event number of the received update into the current version vector entry for the router that generated the update of change events; and

if the least recent event number in the update is not in succession to the event number in the current version vector entry for the router that generated the update of change events, then discarding the received update.

6.   (Original) The method of claim 5 wherein the step of reconciling the database further comprises:

if the update was not discarded in the step of reconciling the current version vector, then for each entry of the received update,

a) determining whether the received entry has expired;

b) if the received entry has expired, then discarding the entry;

c) if the received entry has not expired, then comparing the request identifier of the received entry with the request identifier in the entries in the database;

d) if a matching entry is not found for the received entry, adding the received entry to the database;

e) if a matching entry is found for the received entry, then comparing the application identifier of the received entry with the application identifier of the matching entry;

f) if the application identifiers match, then retaining the entry having a later expiration time in the database; and

g) if the application identifiers do not match, then retaining an entry selected based on a deterministic function applied to a portion of each entry.

7.   (Original) The method of claim 6 wherein the step of retaining an entry based on a deterministic function comprises the steps of applying a function to the application identifiers; and selecting an entry based on the outcome of the function.

8.   (Original) The method of claim 6 wherein the step of retaining an entry based on a deterministic function comprises the steps of applying the deterministic function to the request identifier; and selecting an entry based on the outcome of the deterministic function.

9.   (Original) The method of claim 1 further comprising the step of deleting a binding from the database when the expiration time for the binding has been exceeded .

10.   (Original) The method of claim 1 further comprising the step of sending a request for an update of change events to another router in the plurality; and
        the step of receiving the update further comprises receiving the update in response to the request.

11.   (Original) The method of claim 1 further comprising the steps of:
          periodically generating a first router update of change events;   and,
          transmitting the first router update of change events to at least one other router in the plurality.

12.   (Original) The method of claim 1 further comprising the steps of:
          affirming that an update has been received from each router of the plurality within a predetermined period for each router;
          if an update has not been received from a router within the predetermined period for that router, requesting an update of change events from that router; and
          if an update is received in response to the request,
                reconciling the current version vector according to the received update; and

reconciling the database according to the received update.

13.   (Original) The method of claim 5 wherein the step of reconciling the database further comprises the steps of:

determining from the received update whether the database has a complete record of changes based on the current version vector;

if the database does not have a complete record of changes , requesting a replacement database from a router of the plurality of routers.

14.   (Original) The method of claim 1 further comprising the step of transmitting a copy of the database and the current version vector to another router of the plurality of routers in response to a request from the other router.

15.   (Original) The method of claim 1 wherein the computerized device fails temporarily and recovers, the method further comprising the steps of:

writing the first router change log to a persistent storage device;

sending an update of change events written to the change log in the persistent storage device to other routers in the network;

after recovering from failure, requesting a database and an associated version vector from one of the routers in the plurality;

retaining the received database and associated version vector;

reconciling the received database with the change log from the persistent storage device;   and

updating the received version vector.

16.   (Original) The method of claim 1 wherein the received update includes a version vector and the method of maintaining a current version vector further comprises the step of maintaining the current version vector in a version vector table including past version vectors.

17. (Original) The method of claim 16 further comprising the steps of:

determining from the version vector table whether the database is current based on the version vector table; and

if the database is not current, then requesting missed change events from a second router in the network .

18. (Original) The method of claim 17 wherein each router caches updates received from other routers in the plurality, the method further comprising the step of:

if the router that generated the received update does not respond to the request for missed change events, requesting the missed change events from a second router of the plurality and reconciling the change events into the database and current version vector.

19. (Original) The method of claim 1 wherein the computerized device fails temporarily and recovers, wherein the step of maintaining a current version vector further comprises the steps of:

creating an epoch timestamp from a clock of the computerized device to mark a recovery period;

entering a value pair to the current version vector for the first router, the value pair being an event number and the epoch timestamp; and

the method further comprising the step of after recovery, requesting a database copy and associated version vector from one of the other routers in the plurality.

20. (Original) The method of claim 19 further comprising the steps of :

determining whether a pre-selected time period has passed; and

deleting value pairs before a most recent value pair from the current version vector having timestamps created before the pre-selected time period.

21.   (Original) A system to maintain a client session in a network having a plurality of routers, the network having an application executing at a plurality of replicas, comprising:

     a network interface to receive a request from a client to access the application;

     a storage device to store a database of bindings of requests to replicas where each binding is a record of a request identifier, a replica identifier and a binding expiration time, the storage device to further store a change log and a current version vector associated with the change log where the change log includes records added to the database by the system and the current version vector has an entry for each router in the network, each entry storing a version vector of a particular router; and

     a controller coupled to the interface and the storage device, the controller configured to route the request, if the controller finds a binding matching the request, the controller to route the request to a replica of the plurality according to the binding and to reset the binding expiration time, if the controller does not find a binding matching the request, the controller to add a new record to the database having information from the request to form a binding of the client to a replica of the plurality of replicas.

22.   (Original) The system of claim 21 wherein the network interface is configured to receive an update of change events for the database from another router in the plurality of routers in the network; and the controller is further configured to reconcile the database according to the received update and to update the current version vector in response to reconciling the database.

23.   (Original) The system of claim 21 wherein the controller is configured to transmit periodically, to at least one of the other routers in the plurality, an update of change events and the current version vector.

24.   (Original) A computerized device to maintain a client session in a network having a plurality of such computerized devices, the network having an application executing at a plurality of replicas, comprising:

      means for providing a database of bindings of request identifiers to replicas where each binding is a record having a request identifier, a replica identifier and a binding expiration time, the database associated with a first router of the plurality of routers;

      means for maintaining a change log of records entered into the database, each change log entry having a change event generated by the first router and an event number sequential to an event number of a preceding change event in the change log;

      means for maintaining a current version vector associated with the database and the change log, the current version vector entry for the first router being a most recent event number from the change log, the current version vector entry for each other router being a most recent event number received at the first router from that other router;

      means for receiving an update of change events generated at another router in the plurality;

      means for reconciling the current version vector according to the received update; and

      means for reconciling the database according to the received update such that the client session is maintained.

25.   (Original) A computer program product having a computer-readable medium including computer program logic encoded thereon that, when performed on a computer system having a coupling of a memory, a processor, and at least one communications interface, provides a method for maintaining a client session in a network having an application executing at a plurality of replicas by performing the operations of:

      providing a database of bindings of request identifiers to replicas where each binding is a record having a request identifier, a replica identifier and a binding expiration time, the database associated with a first router of the plurality of routers;

maintaining a change log of records entered into the database, each change log entry having a change event generated by the first router and an event number sequential to an event number of a preceding change event in the change log;

maintaining a current version vector associated with the database and the change log, the current version vector entry for the first router being a most recent event number from the change log, the current version vector entry for each other router being a most recent event number received at the first router from that other router;

receiving an update of change events generated at another router in the plurality;

reconciling the current version vector according to the received update; and

reconciling the database according to the received update such that the client session is maintained.

26.   (Original) A method in a computerized device for maintaining a client session in a network having a plurality of routers, the network having an application executed at a plurality of replicas, comprising the steps of:

providing a database of bindings of requests from clients to replicas where each binding is a record having a request from a client, a replica identifier and a binding expiration time, the database associated with a first router of the plurality of routers;

maintaining a change log of records entered into the database, each change log entry having a change event generated by the first router and an event number sequential to an event number of a preceding change event in the change log;

maintaining a current version vector associated with the database and the change log, the current version vector entry for the first router being a most recent event number from the change log, the current version vector entry for each other router being a most recent event number received at the first router from that other router;

receiving a request from the client to access the application, the request having a client identifier and an application identifier;

comparing data from the request with records in the database;

if the database has a record with a client identifier and an application identifier matching the data from the request,

a) routing the request to a replica of the plurality of replicas according to the matching record; and

b) resetting the binding expiration time of the matching record; and

if the database does not have a record with a client identifier and a replica identifier matching the data from the request,

a) routing the request to one of the plurality of replicas; and

b) entering, into the database a new record forming a binding of the request to one of the replicas.

27.   (Original) The method of claim 26 wherein the router is a DNS server and wherein the application identifier in the request is a domain name and wherein the step of routing comprises mapping the request to an Internet Protocol address of the one replica.